3. In a Word document, describe why you chose the attributes and data types that you chose and why you implemented the relationship(s) between the two models in the way that you did. You may wish to contrast your choices with alternatives and consider the functionality enabled by your choices.

Total 3 marks; 1 mark for attribute explanation, 1 mark for data type explanation and 1 mark for relationship explanation.

* I created 2 models, Course and Student. The attribute info is shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Attribute** | **Data type** | **Explanation** |
| Course | name | CharField | From SPEC: “Each Course has a course name, the name of course coordinator, and size (Maximum number of students).”. I create attributes as SPEC described. Besides, I added 1 more attribute named “current\_size” to record how many students enrolled on this course currently. When current\_size == max\_size, this course can’t accept enrollment from new students.  Datatype: “name” and “coordinator” are characters so use “CharField”. “max\_size” and “current\_size” are used to record number so define them as “IntegerField”. |
| coordinator | CharField |
| max\_size | IntegerField |
| current\_size | IntegerField |
| Student | name | CharField | From SPEC: “Each Student has a name, email address, student id, and date of birth”. I create attributes as SPEC described.  Datatype: “name” contains characters so use “CharField”, “EmailField” is good for “email” attribute. For “stu\_id”, t consists of 8 numbers, but this ID is usually used to be compared not to be calculated, so “CharField” is better. |
| email | EmailField |
| stu\_id | CharField |
| birth\_date | DateField |
| course | ForeignKey |

The relation between “Course” model and “Student” model: From SPEC: “A Student can enroll in only and exactly one Course. However, a Course can have more than one Student.”, I designed an attribute in “Student” model to link “Course” model using “ForeignKey”.

9. In the Word document used for question three, detail which CRUD operations you would restrict to the course coordinator, students and website visitors respectively and justify your decisions.

Total 3 marks: 1 mark for each group and its operations justified.

🡪 CRUD operations are: create, read, update and delete.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Create** | **Read** | **Update** | **Delete** |
| Course coordinator | √ | √ | √ | √ |
| Students | X | √ | √ | X |
| Website visitors | X | √ | X | X |

Like the administrator, the course coordinator has permission to do all CRUD operations.

Students only read all course info and update student information and course enrollment.

Normal visitors only read the basic information and are not allowed to make any changes.

I have created users: admin(admin), stu\_user(1qaz9ol.), course\_user(1qaz9ol.), visitor\_user(1qaz9ol.) to demo this config.

10. Finally, in the same document, discuss two of the benefits of implementing this App within Django, with respect to either security, performance or ease of development. You may provide two benefits from the same category.

Total 1 mark: 0.5 marks for each benefit.

🡪 There are many benefits to using Django to develop.

Benefit 1: The MTV Architecture (Model-Template-View) helps developers to divide a good work breakdown. I can focus on maintaining data structures in the Model, focus on HTML page design in the Template, and use View to connect data and HTML pages.

Benefit 2: URL path handling. It’s very easy to configure the different URLs and their handling method. I don’t need to write a parser to parse the input URL. The response URL is also very easy to set up.

Benefit 3: Django provides a unified interface to access databases of different types. It’s very easy to access the database without writing an SQL statement.

Benefit 4: Django provides an admin interface for the users to maintain backend data conveniently.

Benefit 5: Django provides many middlewares to help developers handle complex events, like sessions, authentication, and security. This will liberate developers and allow them to focus on application development.